UNITED STATES PATENT APPLICATION FOR:

SYSTEM AND METHOD FOR IMPLEMENTING BUSINESS PROCESSES IN A PORTAL

Inventors:

Daryl B. Olander
Richard Feit
Edward O'Neil
Sathyanarayana Giridhar
Jason Howes
Douglas Dew
Thomas A. Cook
Skip Sauls
Rodney McCauley
Michael Blevins
John Shafer
Pal Takacsi-Nagy

CERTIFICATE OF MAILING BY "EXPRESS MAIL"
UNDER 37 C.F.R. §1.10

"Express Mail" mailing label number: EL976798369US
Date of Mailing: 2/25/04

I hereby certify that this correspondence is being deposited with the United States Postal Service, utilizing the "Express Mail Post Office to Addressee" service addressed to: **MAIL STOP PATENT APPLICATION, Commissioner for Patents, Alexandria, VA 22313-1450** and mailed on the above Date of Mailing with the above "Express Mail" mailing label number.

_________________________(Signature)
Name: Tina M. Galdos
Signature Date: 2/25/04

SYSTEM AND METHOD FOR IMPLEMENTING BUSINESS PROCESSES IN A PORTAL

Inventors:

Daryl B. Olander
Richard Feit
Edward O'Neil
Sathyanarayana Giridhar
Jason Howes
Douglas Dew
Thomas A. Cook
Skip Sauls
Rodney McCauley
Michael Blevins
John Shafer
Pal Takacsi-Nagy

COPYRIGHT NOTICE

CLAIM OF PRIORITY

[0002] This application claims priority from the following application, which is hereby incorporated by reference in its entirety:

[0003] SYSTEM AND METHOD FOR IMPLEMENTING BUSINESS PROCESSES IN A PORTAL, U.S. Application No., 60/451,348; Inventors: Daryl Olander et al., filed on February 28, 2003. (Attorney's Docket No. BEAS-1405US0)

CROSS-REFERENCE TO RELATED APPLICATIONS

[0004] This application is related to the following co-pending applications which are each hereby incorporated by reference in their entirety:

[0005] OPEN MARKET COLLABORATION SYSTEM FOR ENTERPRISE WIDE ELECTRONIC COMMERCE, U.S. Application No.

60/183,067, Inventors: Rocky Stewart, et al., filed on February 16, 2000. (Attorney's Docket No. BEAS-1033US1)

[0006] SYSTEMS AND METHODS UTILIZING A WORKFLOW DEFINITION LANGUAGE, U.S. Application No. 60/450,074, Inventor: Pal Takacsi-Nagy, filed on February 25, 2003. (Attorney's Docket No. BEAS-1389US0)

[0007] SYSTEMS AND METHODS FOR CLIENT-SIDE FILTERING OF SUBSCRIBED MESSAGES, U.S. Application No. 60/450,061, Inventors: Mike Blevins, et al., filed on February 25, 2003. (Attorney's Docket No. BEAS-1390US0)

[0008] SYSTEM AND METHOD FOR DYNAMIC DATA BINDING IN DISTRIBUTED APPLICATIONS, U.S. Application No. 60/450,516, Inventor: Edward O'Neil, filed on February 26, 2003. (Attorney's Docket No. BEAS-1448US0)

[0009] SYSTEMS AND METHODS FOR AN EXTENSIBLE CONTROLS ENVIRONMENT, U.S. Application No. 60/451,352, Inventors: Kyle Marvin, et al., filed on February 28, 2003. (Attorney's Docket No. BEAS-1444US0)

[0010] SYSTEM AND METHOD FOR STRUCTURING DISTRIBUTED APPLICATIONS, U.S. Application No. 60/450,226, Inventors: Daryl Olander, et al., filed on February 25, 2003. (Attorney's Docket No. BEAS-1402US0)

[0011] REUSABLE SOFTWARE CONTROLS, U.S. Application No. ____________, Inventors: Kyle Marvin et al., filed on February 17, 2004. (Attorney's Docket No. BEAS-1354US0)

FIELD OF THE DISCLOSURE

[0012] The present invention disclosure relates to a system and method for implementing business processes in a web portal.

BACKGROUND

[0013] Information technology (IT) systems are often implemented as standalone systems, each addressing particular business issues. Enterprises are faced with the task of integrating these disparate systems. Traditionally, islands of individual applications that needed to share data have been integrated in an ad hoc manner, using homegrown integration. Hard-coded and point-to-point solutions attempt to address the problem but instead can create an IT nightmare of innumerable spaghetti-like connections between applications. These problems are further compounded when trying to surface the functionality of more than one IT system in a single web portal programming paradigm. A development framework is needed to simplify and unify integration of disparate IT systems such that their functionality can be surfaced and interact in the context of a web portal.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] **Figure 1** is an illustration of exemplary workflows in one embodiment of the invention.

[0015] **Figure 2** is an illustration of an exemplary system in an embodiment of the invention.

DETAILED DESCRIPTION

[0016] The invention is illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

[0017] Many businesses have adopted the concept of workflows to automate business processes. In one embodiment, a workflow generally refers to a software component that is capable of performing a specific set of tasks. These tasks are typically connected in a way that allows them to be ordered upon the completion of the tasks, which can include work items or other workflows. In a workflow, information such as files, documents, or tasks are passed between system resources according to a set of procedural rules so that the system can act upon the information.

[0018] **Figure 1** is an illustration of exemplary workflows in one embodiment of the invention. Trigger workflow **100** sends an order **102** (which is in the form of an XML document) to main workflow **106**. The order can contain parameters such as a part number and quantity. The main workflow accepts the order and processes it. Processing of the order might entail determining whether or not a sufficient quantity of the desired part is in stock. Part of the processing requires that the main workflow invoke a subsidiary or sub-workflow **108**. Workflows can invoke any number of sub-workflows. Likewise, sub-workflows can invoke other workflows such that invocations of workflows can be nested infinitely. Although not illustrated in **Figure**

1, each workflow can execute on the same or different computing device. In addition, a given workflow can be distributed across a plurality of computing devices.

[0019] Workflows can be initiated and interacted with via message brokers. A message broker can operate in a server cluster and can allow any server in that cluster to publish to and subscribe to a channel. A server subscribing to a channel can set up a message filter such that XQuery expressions can be used to filter against the messages arriving on the channel. XQuery is a standard query language for processing XML data and data whose structure is similar to XML. A user can write or create an XQuery expression that can extract a portion of the (XML) document payload. The server can then compare the extracted fragment against a filter value for that channel, and if the XML fragment matches the filter value then the server receives the message.

[0020] The logic for interaction with message brokers can be encapsulated in one or more controls. Controls can be surfaced in integrated development environments (IDEs). Controls allow a software developer to "plug in" a -- sometimes complex -- piece of functionality, without the need to understand the details. For example, a developer might drag a graphical representation of a business control off a palette in an IDE and "drop" it onto a graphical canvas wherein it can be connected to other controls or objects in a flow of control. Typically, controls expose one or more interfaces that allow the software developer to access the functionality packaged within the control. Furthermore, controls can communicate with each other, thereby allowing disparate IT workflows to interoperate and exchange information.

[0021] Controls can be utilized in portal web pages (or page groups). Page groups include classes that contain user interface control logic. User interface control logic is logic that implements navigation decisions, that flows data into and out of pages and that invokes back-end business logic via calls to controls. Page groups enable developers to simplify the organization of web applications. Developers may use page groups to organize the files of a web application into small, focused units. For example, a developer might organize the files of the human resources portion of a large web application into benefitsWizard, help, hiringWizard, login, payAdjustment Wizard, stockPurchaseWizard and vacationWizard page groups. Page groups are the main part of a web application programming model that enables architects and developers to design and implement web apps using an easy-to-understand, easy-tomaintain, fill-in-the-blanks pattern.

[0022] Controls make it easy to access enterprise resources, such as databases, file systems, Enterprise Java Beans, and so on, from within your application. The control handles the work of connecting to the enterprise resource so that programmers can focus on business process logic. When a resource is accessed through a control, the interaction with the resource can be greatly simplified since the underlying control implementation takes care of most of the details. Users need only add an instance of a control to a business process and then invoke its methods. All controls can expose Java interfaces that can be invoked directly from a business process.

[0023] Control nodes can be added to a business process to represent points in the business process at which you design interactions with resources via controls. Control Send nodes represent points in business processes at which the business processes send messages to resources via controls. Control Receive nodes represent points in business processes at which the business processes receive asynchronous messages from resources via controls. Business processes can wait at these nodes until they receive a message from the specified control. Control Send with Return nodes can handle synchronous exchange of messages between business process and resources via controls.

[0024] Some controls are transactional. This means that the control is able to participate in transactions within a business process. Whether or not a control is transactional depends on both the underlying resource and the specific control implementation. Also, transactional behavior differs depending on whether the control call is synchronous or asynchronous. If the control and associated resource are transactional, the resource participates in the current process transaction. If the control and associated resource are not transactional, changes to the resource occur outside the scope of the current transaction and changes are not rolled back in case of failure.

[0025] For asynchronous control calls, the process transaction is not propagated to the resource. Asynchronous control calls are buffered by default. Asynchronous call to the resource are not enqueued until the transaction is committed. On rollback, asynchronous messages are de-queued. A Process control is a special case, since it involves processes calling subprocesses. For synchronous operations the transaction is always propagated to the subprocess. An un-handled exception in a subprocess causes the shared transaction to be marked as rollback only. In this case, both the subprocess and the calling process are rolled back. The subprocess can run in its own transaction.

[0026] In one embodiment and by way of example, the following controls are transactional:

- Application View (if JCA adapter is transactional)
- Message Broker
- Process (see the previously listed qualifications)
- TPM
- WLI JMS
- Worklist

[0027] By way of example, the following integration controls are not transactional:

- File
- Email
- Service Broker
- RosettaNet
- ebXML

[0028] The system can support custom controls and is flexible, supporting a wide variety of uses for controls. Controls can contain business logic so as to keep it separate from other application code, or which may be reused. Java controls can provide access to resources such as databases or other resources and collect logic that coordinates multiple actions, such as those that involve multiple database queries, calls to Enterprise JavaBeans (with the EJB control), and so on. A control can participate in the implicit transaction of a conversational container, such as a web service that is conversational.

[0029] **Figure 2** is an illustration of an exemplary system in an embodiment of the invention. Although this diagram depicts objects/processes as logically separate, such depiction is merely for illustrative purposes. It will be apparent to those skilled in the art that the objects/processes portrayed in this figure can be arbitrarily combined or divided into separate software, firmware and/or hardware components. Furthermore, it will also be apparent to those skilled in the art that such objects/processes, regardless of how they are combined or divided, can execute on the same computing device or can be distributed among different computing devices connected by one or more networks or other suitable communication means.

[0030] Referring to **Figure 2**, client tier **200** might include a web browser to render HTML and execute Applets. The client tier interacts with a page group **202** in the web layer **204**. The page group can include one or more controls (**206-208**). Control **208** is capable of communicating with message broker **210** in the business logic layer **214** through which it can invoke workflow **212**. In one embodiment, a message broker can be a JMS message queue. The workflow in turn may invoke other workflows and/or communicate with the back-end tier **216**. Thus, workflows, controls and page groups can be assembled into a powerful, highly leveraged framework for integrating IT systems. Workflows can be accessed through controls. Page groups can integrate controls into a portal and thereby provide a user interface.

[0031] One embodiment may be implemented using a conventional general purpose or a specialized digital computer or microprocessor(s) programmed according to the teachings of the present disclosure, as will be apparent to those skilled in the computer art. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art. The invention may also be implemented by the preparation of integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

[0032] One embodiment includes a computer program product which is a storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the features presented herein. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

[0033] Stored on any one of the computer readable medium (media), the present invention includes software for controlling both the hardware of the general purpose/specialized computer or microprocessor, and for enabling the computer or microprocessor to interact with a human user or other mechanism utilizing the results of the present invention. Such software may include, but is not limited to, device drivers, operating systems, execution environments/containers, and user applications.

[0034] The foregoing description of the preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to the practitioner skilled in the art. Embodiments were chosen and described in order to best describe the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention, the various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.